

Первая нормальная форма

Определение: «Все атрибуты отношения атомарны».

Пояснение человеческим языком: «Мы [почти] никогда на уровне СУБД не анализируем части полей таблицы».

users

u_id (PK)	u_email
1	pupkin@gmail.com
2	ivanov@mail.ru

Получим 1НФ:

users

u_id (PK)	u_email_user	u_email_domain
1	pupkin	gmail.com
2	ivanov	mail.ru

При условии, что нам приходится часто выполнять задачи вида «разослать сообщения всем, у кого почта на Gmail'е», надо атомизировать поле «email», разбив его на два отдельных поля (так, чтобы хранить доменное имя в явном отдельном виде).

Вторая нормальная форма

Определение: «1НФ + любой неключевой атрибут функционально полно зависит от ПК».

Пояснение человеческим языком: «Не существует полей, не входящих в состав ПК, и при этом зависящих от части ПК».

Быстрое формальное решение: добавьте суррогатный ПК из одного поля, и теперь никакое неключевое поле не сможет зависеть от **части** ПК, т.к. у ПК из одного поля нет частей.

Наличие ноутбука зависит от части ПК (если принято, что у начальника ноутбук есть, а у подчинённого – нет):

employees

e_name (PK)	e_position (PK)	e_has_notebook
Иванов	Начальник	Да
Иванов	Подчинённый	Нет
Петров	Подчинённый	Нет

Получим 2НФ:

employees

e_id (PK)	e_name	e_position
1	Иванов	1
2	Иванов	2
3	Петров	2

positions

p_id (PK)	p_name	p_notebook
1	Начальник	Да
2	Подчинённый	Нет

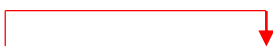
Третья нормальная форма

Определение: «2НФ + любой неключевой атрибут нетранзитивно зависит от ПК».

Пояснение человеческим языком: «Любой неключевой атрибут должен зависеть только от ПК, а не от других неключевых атрибутов».

Быстрое формальное решение: храните в одном отношении только те свойства описываемого объекта, значения которых не будут многократно дублироваться у разных объектов (в примере ниже многократно будут дублироваться телефоны офисов – столько раз, сколько в офисе будет сотрудников).

employees



e_id (PK)	e_name	e_office	e_office_phone
1	Иванов	Офис-1	123-45-67
2	Иванов	Офис-2	555-66-77
3	Петров	Офис-2	555-66-77

Получим 3НФ:

employees

e_id (PK)	e_name	e_office
1	Иванов	1
2	Иванов	2
3	Петров	2

offices

o_id (PK)	o_name	o_phone
1	Офис-1	123-45-67
2	Офис-2	555-66-77

Важно: тут мы допускаем, что у офиса есть только один телефон.

Нормальная форма Бойса-Кодда

Определение: «3НФ + детерминанты всех функциональных зависимостей являются потенциальными ключами».

Пояснение человеческим языком: «Не должно быть потенциальных ключей, имеющих общие атрибуты (пересечения)».

education

e_passport	e_subject	e_code	e_mark
AA111111	Химия	S001	10
AA222222	Физика	S002	9
AA222222	Математика	S002	10



Потенциальные ключи: {e_passport, e_subject} и {e_subject, e_code}.

Вся совокупность полей {e_passport, e_subject, e_code} не является потенциальным ключом, т.к. в нём есть избыточность.

Общий атрибут (пересечение потенциальных ключей): e_subject.

Имеются две ФЗ ($e_passport \rightarrow e_code$, $e_code \rightarrow e_passport$), детерминанты которых не являются потенциальными ключами.

students

s_code (PK)	s_passport
S001	AA111111
S002	AA222222

marks

m_subject (PK)	m_code (PfK)	m_mark
Химия	S001	10
Физика	S002	9
Математика	S002	10

Четвёртая нормальная форма

Определение: «НФБК + отсутствуют нетривиальные многозначные зависимости».

Пояснение человеческим языком: «Все зависимости должны быть функциональными от ключа».

Перечень ограничений предметной области:

- Каждый абитуриент имеет право сдавать экзамены на несколько факультетов одновременно.
- Каждый факультет имеет свой список сдаваемых предметов.
- Один и тот же предмет может сдаваться на нескольких факультетах.
- Абитуриент обязан сдавать все предметы, указанные для факультета, на который он поступает, несмотря на то, что он, может быть, уже сдавал такие же предметы на другом факультете.

education

e_applicant (PK)	e_faculty (PK)	e_subject (PK)
Иванов	Математический	Математика
Иванов	Математический	Информатика
Иванов	Физический	Математика
Иванов	Физический	Физика
Петров	Математический	Математика
Петров	Математический	Информатика

Проблемы:

- Аномалия вставки. При добавлении нового абитуриента на факультет, нужно вставить несколько записей вида «Абитуриент-Факультет-Предмет» для всех предметов этого факультета. При добавлении нового предмета на факультет, надо всем абитуриентам этого факультета добавить этот новый предмет.
- Аномалия удаления. При удалении абитуриента с факультета надо удалить все записи вида «Абитуриент-Факультет-Предмет» (их будет столько, сколько на этом факультете сдаётся предметов). При удалении предмета с факультета надо будет удалить его у всех абитуриентов этого факультета.

Зависимости:

- Многозначная нетривиальная: $e_applicant \twoheadrightarrow e_faculty \mid e_subject$
- Функциональные: нет ни одной (между полями $e_applicant$, $e_faculty$, $e_subject$ нигде нет функциональной зависимости).

Решение («научно верное» (пропадут аномалии), но выглядящее безумно):

applications

e_applicant (PK)	e_faculty (PK)
Иванов	Математический
Иванов	Физический
Петров	Математический

examination

e_faculty (PK)	e_subject (PK)
Математический	Математика
Математический	Информатика
Физический	Математика
Физический	Физика

Решение («как это будет в реальности»):

applicants

a_id (PK)	a_name
1	Иванов
2	Петров

a_m2m_f

a_id (PfK)	f_id (PfK)
1	1
1	2
2	1

faculties

f_id (PK)	f_name
1	Математический
2	Физический

f_m2m_s

f_id (PfK)	s_id (PfK)
1	1
1	2
2	1
2	3

subjects

s_id (PK)	s_name
1	Математика
2	Информатика
3	Физика

Пятая нормальная форма

Определение: «4НФ + любая зависимость соединения является тривиальной (т.е. нет нетривиальных зависимостей соединения)».

Пояснение человеческим языком: «Если отношение разложить на его проекции и “продожинить” их назад, должно получиться в точности исходное отношение. При нарушении 5НФ в полученном отношении получаются записи, которых раньше там не было».

something

s_device	s_os	s_version
Computer	Windows	Server
Computer	Linux	User
Phone	Windows	User
Computer	Windows	User

device_os

s_device	s_os
Computer	Windows
Computer	Linux
Phone	Windows
Computer	Windows

device_version

s_device	s_version
Computer	Server
Computer	User
Phone	User
Computer	User

os_version

s_os	s_version
Windows	Server
Linux	User
Windows	User
Windows	User

something_new

s_device	s_os	s_version
Computer	Linux	Server
Computer	Windows	Server
Computer	Linux	User
Phone	Windows	User
Computer	Windows	User

← Этой записи раньше не было!

Решение: совершенно аналогично 4НФ «резать» до тех пор, пока не получится набор отношений, не подверженный такому неожиданному эффекту.

Доменно-ключевая нормальная форма

Определение: «5НФ + для атрибутов указаны типы и домены + обозначены потенциальные ключи».

Пояснение человеческим языком: «Надо использовать правильные типы данных, для перечислений/множеств использовать ENUM/SET или CHECK, на потенциальных ключах строить индексы, между таблицами явно проводить связи».

events

e_id (PK)	e_date	e_day
1	02.10.2010	Суббота
2	03.11.2011	Четверг
3	04.12.2012	Вторник

events

e_id (PK)	e_date	e_day (FK)
1	02.10.2010	6
2	03.11.2011	4
3	04.12.2012	2

days

d_id (PK)	d_name
1	Понедельник
2	Вторник
3	Среда
4	Четверг
5	Пятница
6	Суббота
7	Воскресенье

Здесь можно ещё сделать ENUM или CHECK на значение названий дней.

Шестая нормальная форма

Определение: «5НФ/ДКНФ + дальнейшая декомпозиция невозможна без потерь данных».

Пояснение человеческим языком: «5НФ + декомпозиция не приводит к потере временных данных (сама 6НФ вводилась именно для хронологических хранилищ и операций с ними)».

life

l_id	l_period	l_position	l_address
111	01-01-2005:31-12-2007	Сторож	пр. Проспектный 111, 999
111	01-01-2008:31-12-2009	Охранник	пр. Проспектный 111, 999
111	01-01-2010:31-12-2011	Охранник	ул. Уличная, 5-12

Конечная декомпозиция будет выглядеть так:

career

c_id (PK)	c_period (PK)	c_position
111	01-01-2005:31-12-2007	Сторож
111	01-01-2008:31-12-2011	Охранник

residence

r_id (PK)	r_period (PK)	r_address
111	01-01-2005:31-12-2009	пр. Проспектный 111, 999
111	01-01-2010:31-12-2011	ул. Уличная, 5-12